Session Report

Test & Feedback

Started On:	**1/1/2023 07:24 PM**
Exported On:	**1/1/2023 08:22 PM**
Bugs Filed:	**3**
Time Zone:	**Eastern European Standard Time**

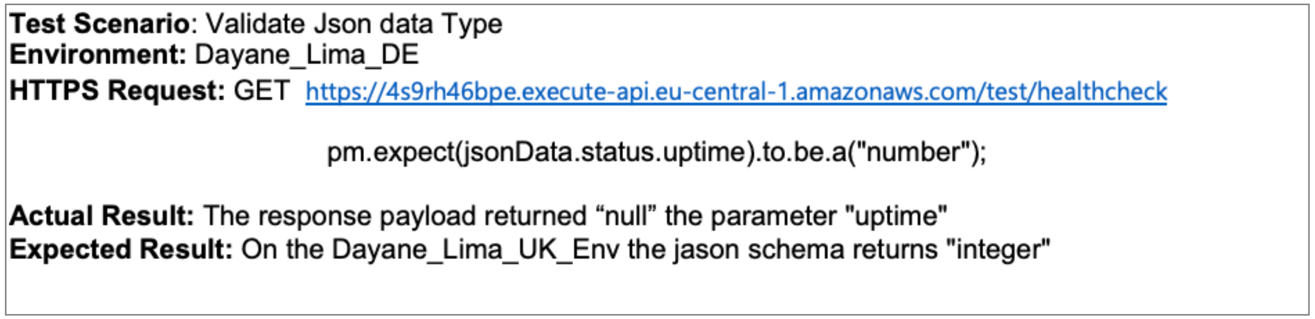
This report was generated using Test & Feedback extension for Azure DevOps Services / Team Foundation Server in standalone mode (with limited functionality). Connect to Azure DevOps for the full featured version, learn more - Test & Feedback Marketplace page.

Summary of bugs filed

Bug 1	1/1/2023 07:53 PM	Dayane_Lima_DE_Env: Paramenter "uptime" Null
Bug 2	1/1/2023 08:06 PM	Dayane_Lima_UK: Misspelling in the "status" parameter
Bug 3	1/1/2023 08:14 PM	Dayane_Lima_UK: The health" is "UNSTABLE”

Bug 1 1/1/2023 07:53 PM

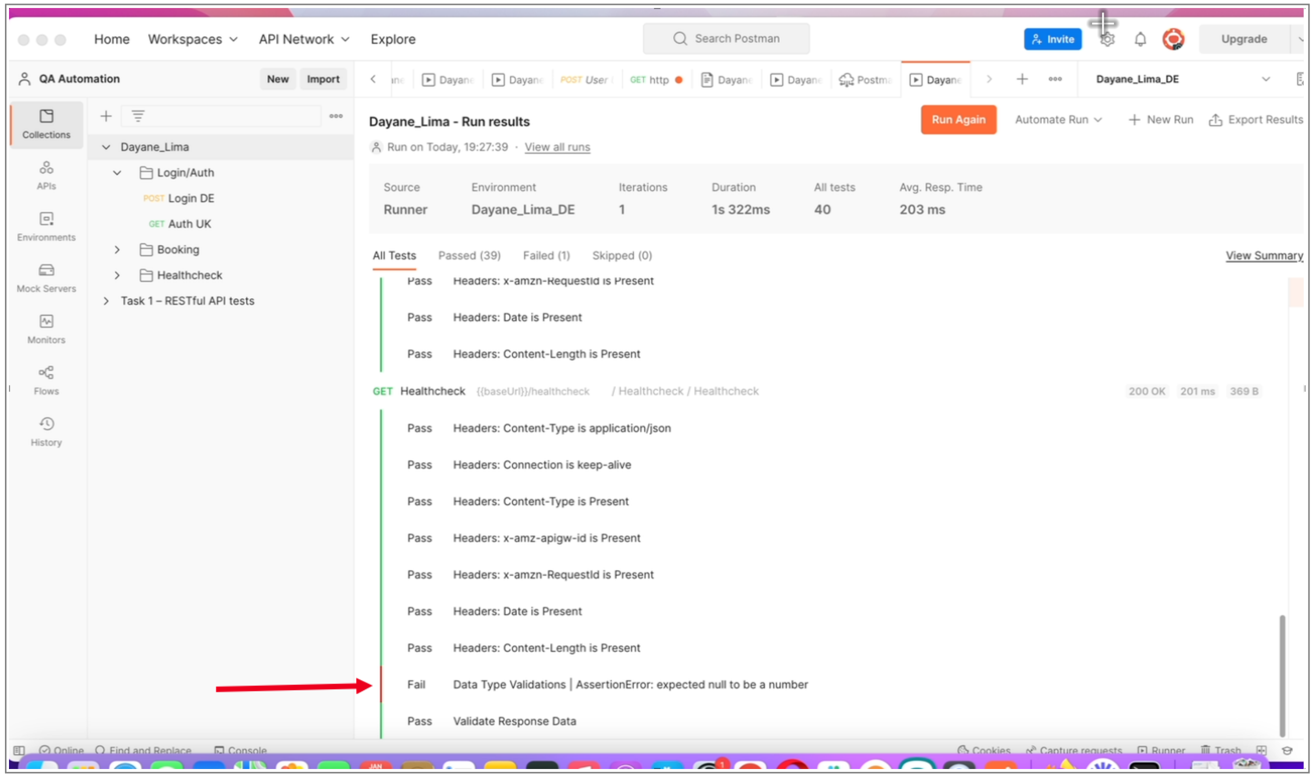
Dayane_Lima_DE_Env: Paramenter "uptime" Null



Test Scenario: Validate Json data Type
Environment: Dayane_Lima_DE
HTTPS Request: GET https://4s9rh46bpe.execute-api.eu-central-1.amazonaws.com/test/healthcheck

```
pm.expect(jsonData.status.uptime).to.be.a("number");
```

Actual Result: The response payload returned “null” the parameter "uptime"
Expected Result: On the Dayane_Lima_UK_Env the jason schema returns "integer"



Dayane_Lima - Run results

Run on Today, 19:27:39 · View all runs

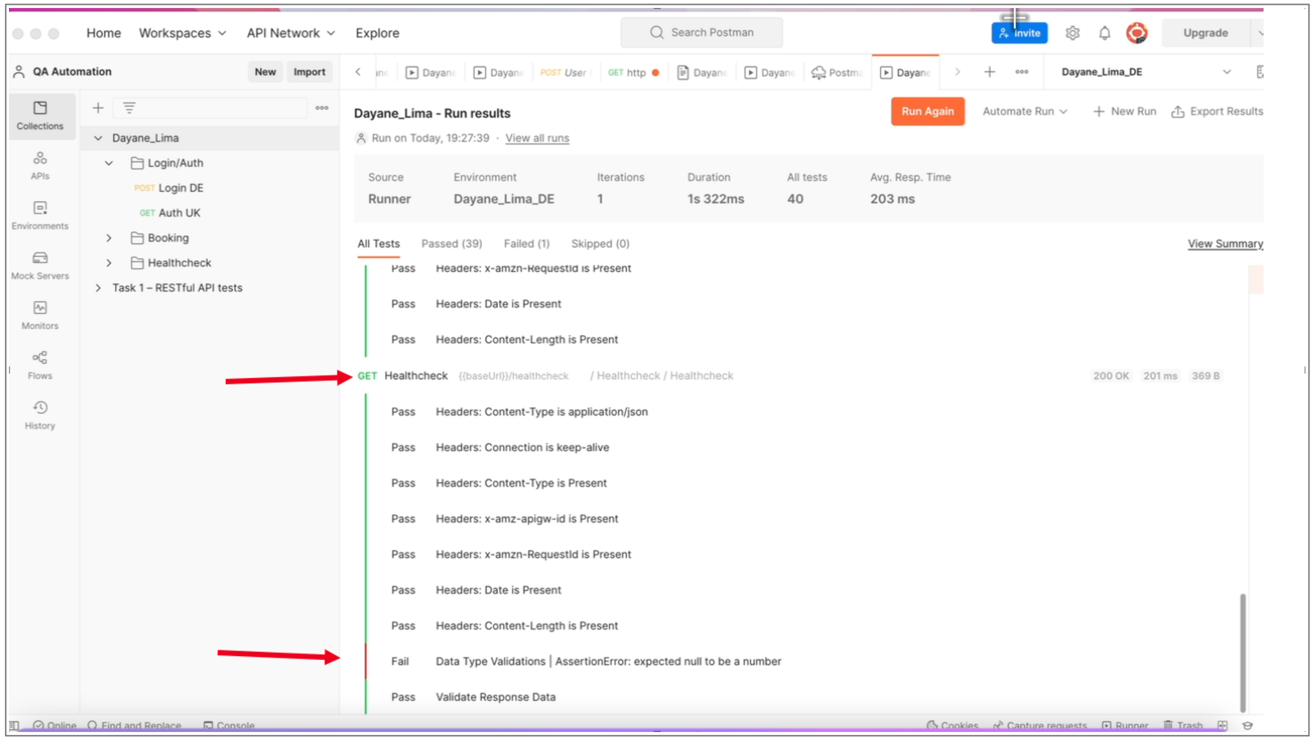
Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	Dayane_Lima_DE	1	1s 322ms	40	203 ms

All Tests   Passed (39)   Failed (1)   Skipped (0)

```
Pass   Headers: x-amzn-RequestId is Present
Pass   Headers: Date is Present
Pass   Headers: Content-Length is Present

GET Healthcheck  {{baseUrl}}/healthcheck   / Healthcheck / Healthcheck    200 OK   201 ms   369 B

Pass   Headers: Content-Type is application/json
Pass   Headers: Connection is keep-alive
Pass   Headers: Content-Type is Present
Pass   Headers: x-amz-apigw-id is Present
Pass   Headers: x-amzn-RequestId is Present
Pass   Headers: Date is Present
Pass   Headers: Content-Length is Present
Fail   Data Type Validations | AssertionError: expected null to be a number
Pass   Validate Response Data
```



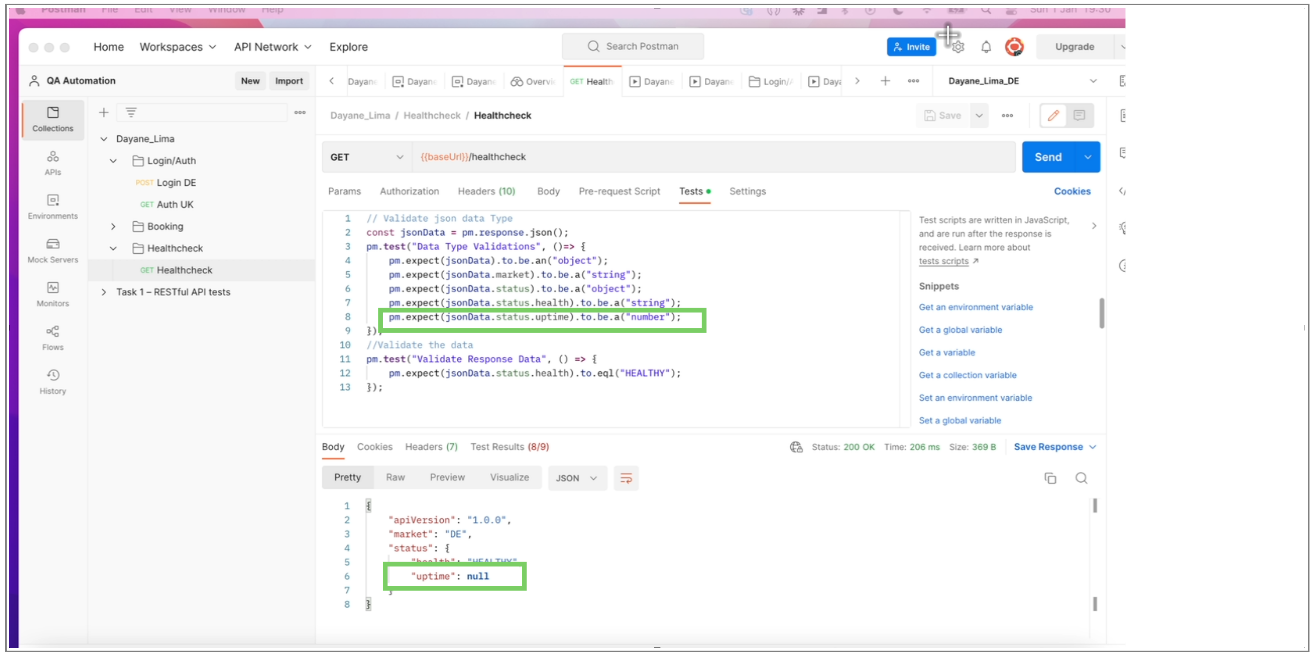
Dayane_Lima - Run results

Run on Today, 19:27:39

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	Dayane_Lima_DE	1	1s 322ms	40	203 ms

All Tests Passed (39) Failed (1) Skipped (0)

```
Pass   Headers: x-amzn-RequestId is Present
Pass   Headers: Date is Present
Pass   Headers: Content-Length is Present
GET Healthcheck  {{baseUrl}}/healthcheck  / Healthcheck / Healthcheck    200 OK  201 ms  369 B
Pass   Headers: Content-Type is application/json
Pass   Headers: Connection is keep-alive
Pass   Headers: Content-Type is Present
Pass   Headers: x-amz-apigw-id is Present
Pass   Headers: x-amzn-RequestId is Present
Pass   Headers: Date is Present
Pass   Headers: Content-Length is Present
Fail   Data Type Validations | AssertionError: expected null to be a number
Pass   Validate Response Data
```



Dayane_Lima / Healthcheck / Healthcheck

GET {{baseUrl}}/healthcheck

Params Authorization Headers (10) Body Pre-request Script Tests Settings

```javascript
// Validate json data Type
const jsonData = pm.response.json();
pm.test("Data Type Validations", ()=> {
    pm.expect(jsonData).to.be.an("object");
    pm.expect(jsonData.market).to.be.a("string");
    pm.expect(jsonData.status).to.be.a("object");
    pm.expect(jsonData.status.health).to.be.a("string");
    pm.expect(jsonData.status.uptime).to.be.a("number");
});
//Validate the data
pm.test("Validate Response Data", () => {
    pm.expect(jsonData.status.health).to.eql("HEALTHY");
});
```

Test scripts are written in JavaScript, and are run after the response is received. Learn more about tests scripts ↗

Body Cookies Headers (7) Test Results (8/9) Status: 200 OK Time: 206 ms Size: 369 B

```json
{
    "apiVersion": "1.0.0",
    "market": "DE",
    "status": {
        "health": "HEALTHY",
        "uptime": null
    }
}
```

Json Schema

```
{
 "$schema": "http://json-schema.org/draft-04/schema#",
 "type": "object",
 "properties": {
  "apiVersion": {
   "type": "string"
  },
  "market": {
   "type": "string"
  },
  "status": {
   "type": "object",
   "properties": {
    "health": {
     "type": "string"
    },
    "uptime": {
     "type": "null"
    }
   },
   "required": [
    "health",
    "uptime"
   ]
  }
 },
 "required": [
  "apiVersion",
  "market",
  "status"
 ]
}
```

Bug 2 1/1/2023 08:06 PM

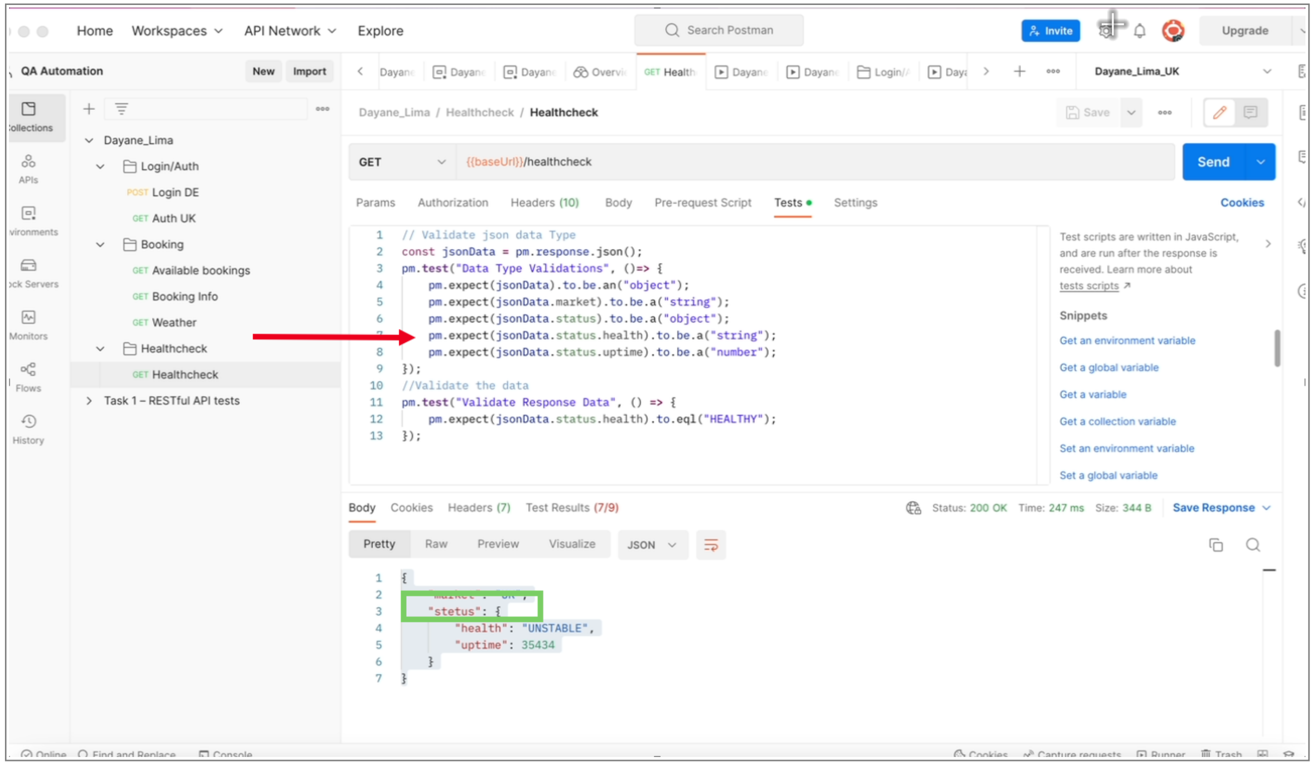
Dayane_Lima_UK: Misspelling in the "status" parameter

Test Scenario: Validate Json Response Body
Environment: Dayane_Lima_UK
HTTPS Request: GET https://4s9rh46bpe.execute-api.eu-central-1.amazonaws.com/test/healthcheck

Failed test:

```
    pm.expect(jsonData.status).to.be.a("object");
  pm.expect(jsonData.status.health).to.be.a("string");
  pm.expect(jsonData.status.uptime).to.be.a("integer");
```

Actual Result: On the response payload - Misspelling in the "stetus" parameter
Expected Result: Response payload must display “Status”



GET {{baseUrl}}/healthcheck

```
// Validate json data Type
const jsonData = pm.response.json();
pm.test("Data Type Validations", ()=> {
    pm.expect(jsonData).to.be.an("object");
    pm.expect(jsonData.market).to.be.a("string");
    pm.expect(jsonData.status).to.be.a("object");
    pm.expect(jsonData.status.health).to.be.a("string");
    pm.expect(jsonData.status.uptime).to.be.a("number");
});
//Validate the data
pm.test("Validate Response Data", () => {
    pm.expect(jsonData.status.health).to.eql("HEALTHY");
});
```

Test scripts are written in JavaScript, and are run after the response is received. Learn more about tests scripts ↗

```
{
    "market": "UK",
    "stetus": {
        "health": "UNSTABLE",
        "uptime": 35434
    }
}
```

Bug 3 1/1/2023 08:14 PM

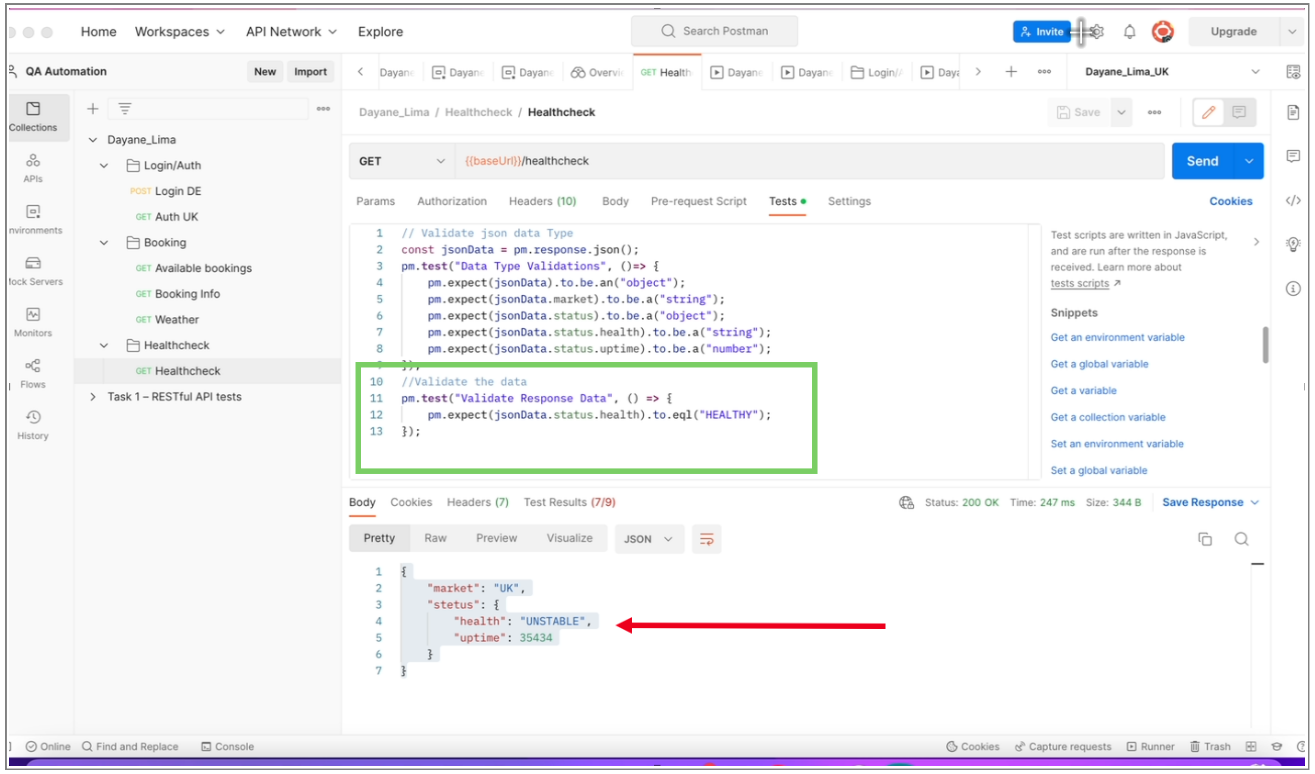
Dayane_Lima_UK: The health" is "UNSTABLE”

Test Scenario: Validate Json Response Body
Environment: Dayane_Lima_UK
HTTPS Request: GET https://4s9rh46bpe.execute-api.eu-central-1.amazonaws.com/test/healthcheck

Failed test:

```
        pm.test("Validate Response Data", () => {
            pm.expect(jsonData.status.health).to.eql("HEALTHY");
```

Actual Result: The The health" is "UNSTABLE"
Expected Result: The Dayane_Lima_UK must be "health"



GET {{baseUrl}}/healthcheck

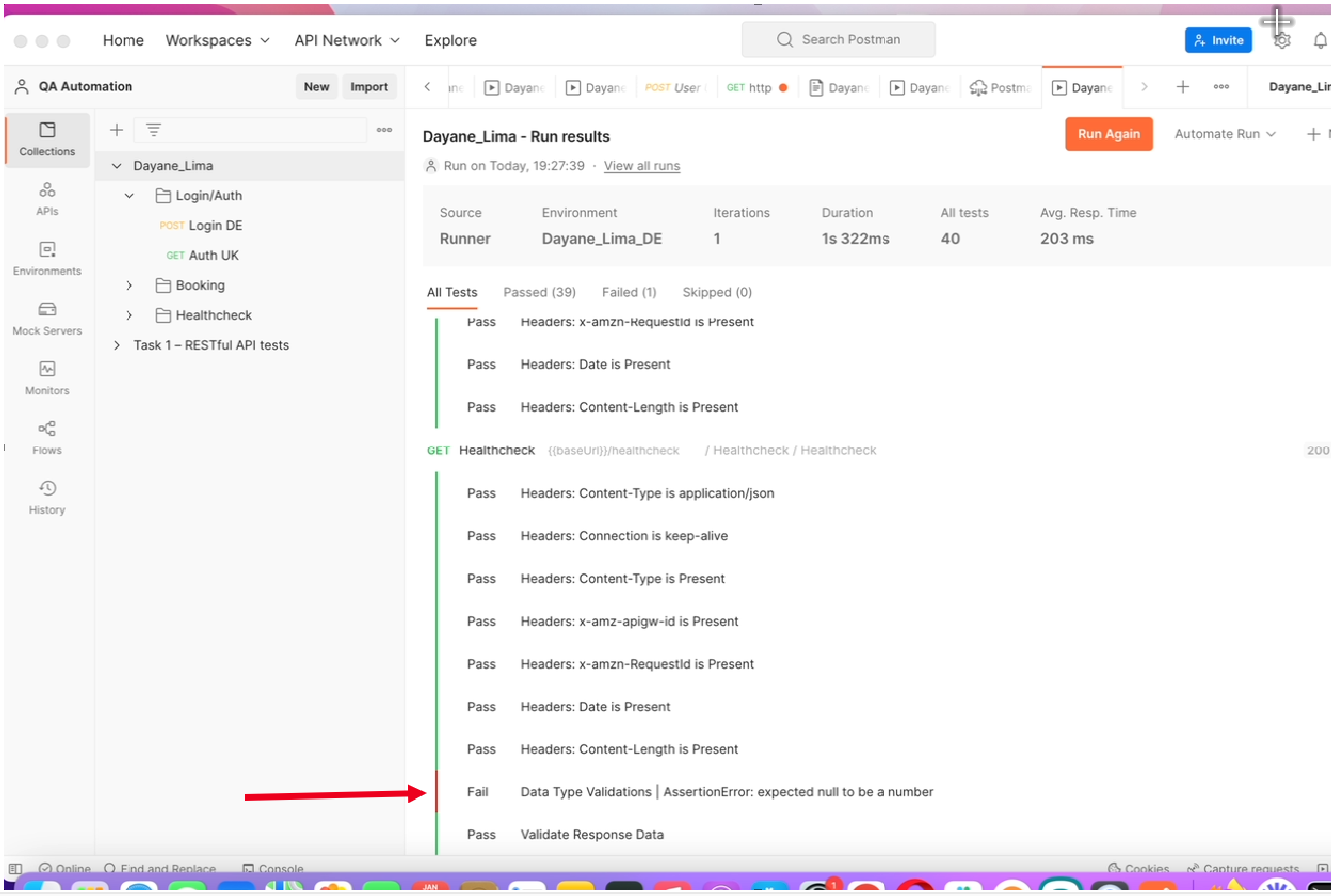
```
// Validate json data Type
const jsonData = pm.response.json();
pm.test("Data Type Validations", ()=> {
    pm.expect(jsonData).to.be.an("object");
    pm.expect(jsonData.market).to.be.a("string");
    pm.expect(jsonData.status).to.be.a("object");
    pm.expect(jsonData.status.health).to.be.a("string");
    pm.expect(jsonData.status.uptime).to.be.a("number");
});
//Validate the data
pm.test("Validate Response Data", () => {
    pm.expect(jsonData.status.health).to.eql("HEALTHY");
});
```

Test Results (7/9) Status: 200 OK Time: 247 ms Size: 344 B

```
{
    "market": "UK",
    "stetus": {
        "health": "UNSTABLE",
        "uptime": 35434
    }
}
```

Session attachments

Screenshot-1.png 1/1/2023 07:28 PM



Dayane_Lima - Run results

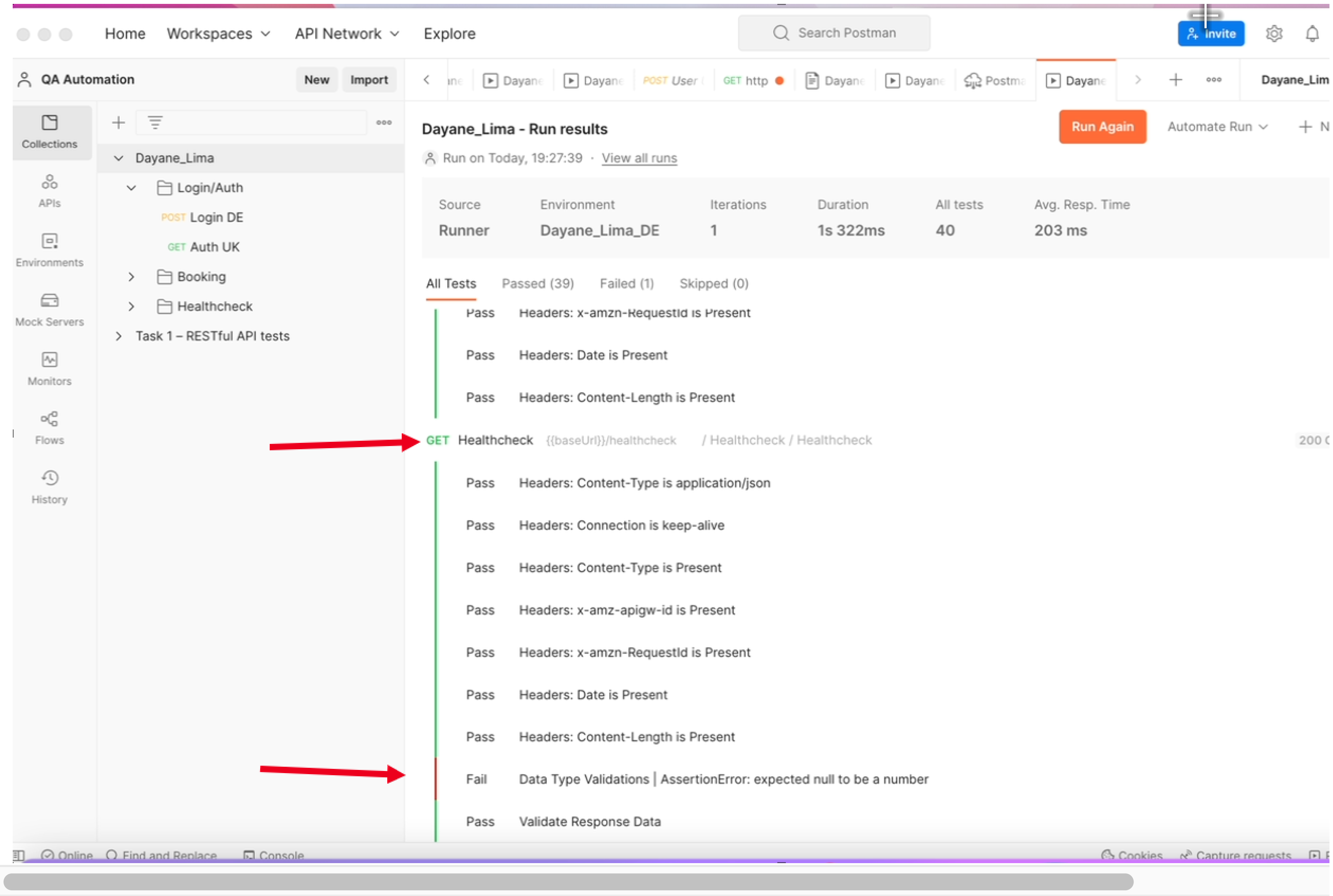
Run on Today, 19:27:39 · View all runs

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	Dayane_Lima_DE	1	1s 322ms	40	203 ms

All Tests Passed (39) Failed (1) Skipped (0)

```
Pass   Headers: x-amzn-RequestId is Present
Pass   Headers: Date is Present
Pass   Headers: Content-Length is Present
GET Healthcheck {{baseUrl}}/healthcheck / Healthcheck / Healthcheck     200
Pass   Headers: Content-Type is application/json
Pass   Headers: Connection is keep-alive
Pass   Headers: Content-Type is Present
Pass   Headers: x-amz-apigw-id is Present
Pass   Headers: x-amzn-RequestId is Present
Pass   Headers: Date is Present
Pass   Headers: Content-Length is Present
Fail   Data Type Validations | AssertionError: expected null to be a number
Pass   Validate Response Data
```

Screenshot-2.png 1/1/2023 07:29 PM



Dayane_Lima - Run results

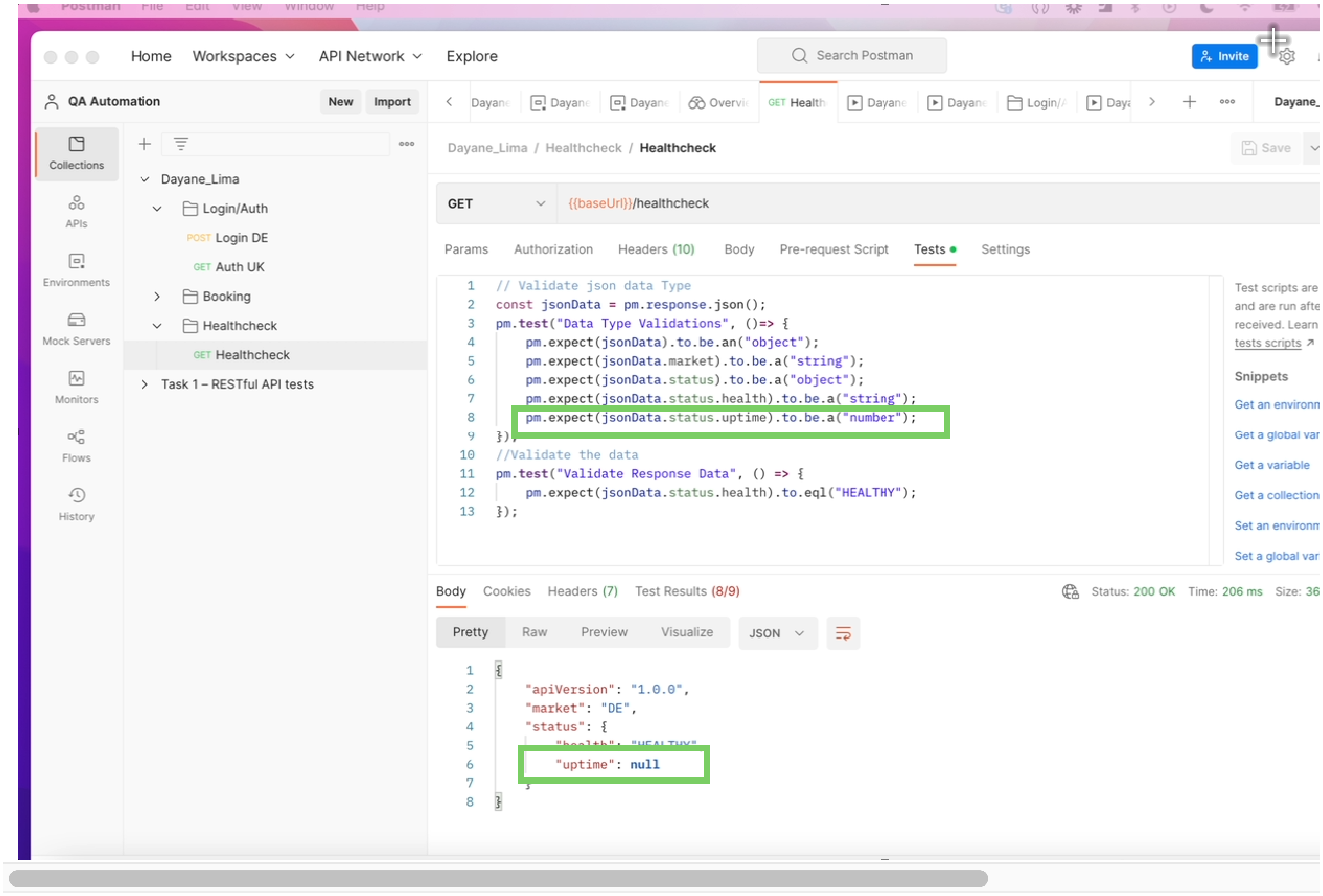
Run on Today, 19:27:39 · View all runs

Source	Environment	Iterations	Duration	All tests	Avg. Resp. Time
Runner	Dayane_Lima_DE	1	1s 322ms	40	203 ms

All Tests Passed (39) Failed (1) Skipped (0)

```
Pass   Headers: x-amzn-RequestId is Present
Pass   Headers: Date is Present
Pass   Headers: Content-Length is Present
GET Healthcheck  {{baseUrl}}/healthcheck   / Healthcheck / Healthcheck   200
Pass   Headers: Content-Type is application/json
Pass   Headers: Connection is keep-alive
Pass   Headers: Content-Type is Present
Pass   Headers: x-amz-apigw-id is Present
Pass   Headers: x-amzn-RequestId is Present
Pass   Headers: Date is Present
Pass   Headers: Content-Length is Present
Fail   Data Type Validations | AssertionError: expected null to be a number
Pass   Validate Response Data
```

Screenshot-3.png 1/1/2023 07:31 PM



GET {{baseUrl}}/healthcheck

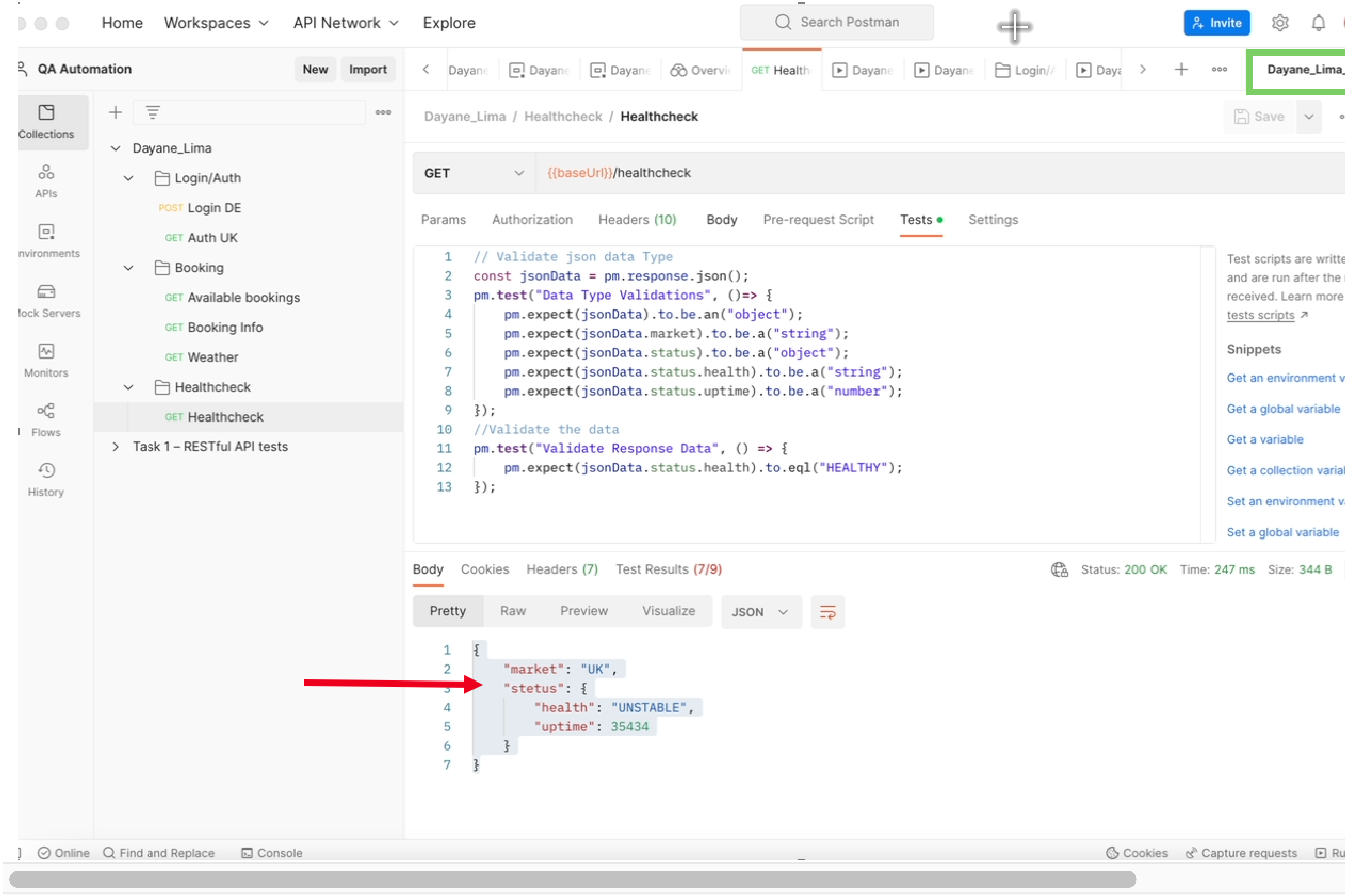
```
// Validate json data Type
const jsonData = pm.response.json();
pm.test("Data Type Validations", ()=> {
    pm.expect(jsonData).to.be.an("object");
    pm.expect(jsonData.market).to.be.a("string");
    pm.expect(jsonData.status).to.be.a("object");
    pm.expect(jsonData.status.health).to.be.a("string");
    pm.expect(jsonData.status.uptime).to.be.a("number");
});
//Validate the data
pm.test("Validate Response Data", () => {
    pm.expect(jsonData.status.health).to.eql("HEALTHY");
});
```

```
{
    "apiVersion": "1.0.0",
    "market": "DE",
    "status": {
        "health": "HEALTHY",
        "uptime": null
    }
}
```

Note-1 1/1/2023 07:34 PM

Json Schema { "$schema": "http://json-schema.org/draft-04/schema#", "type": "object", "properties": { "apiVersion": { "type": "string" }, "market": { "type": "string" }, "status": { "type": "object", "properties": { "health": { "type": "string" }, "uptime": { "type": "null" } }, "required": ["health", "uptime"] } }, "required": ["apiVersion", "market", "status"] }

Screenshot-7.png 1/1/2023 07:54 PM



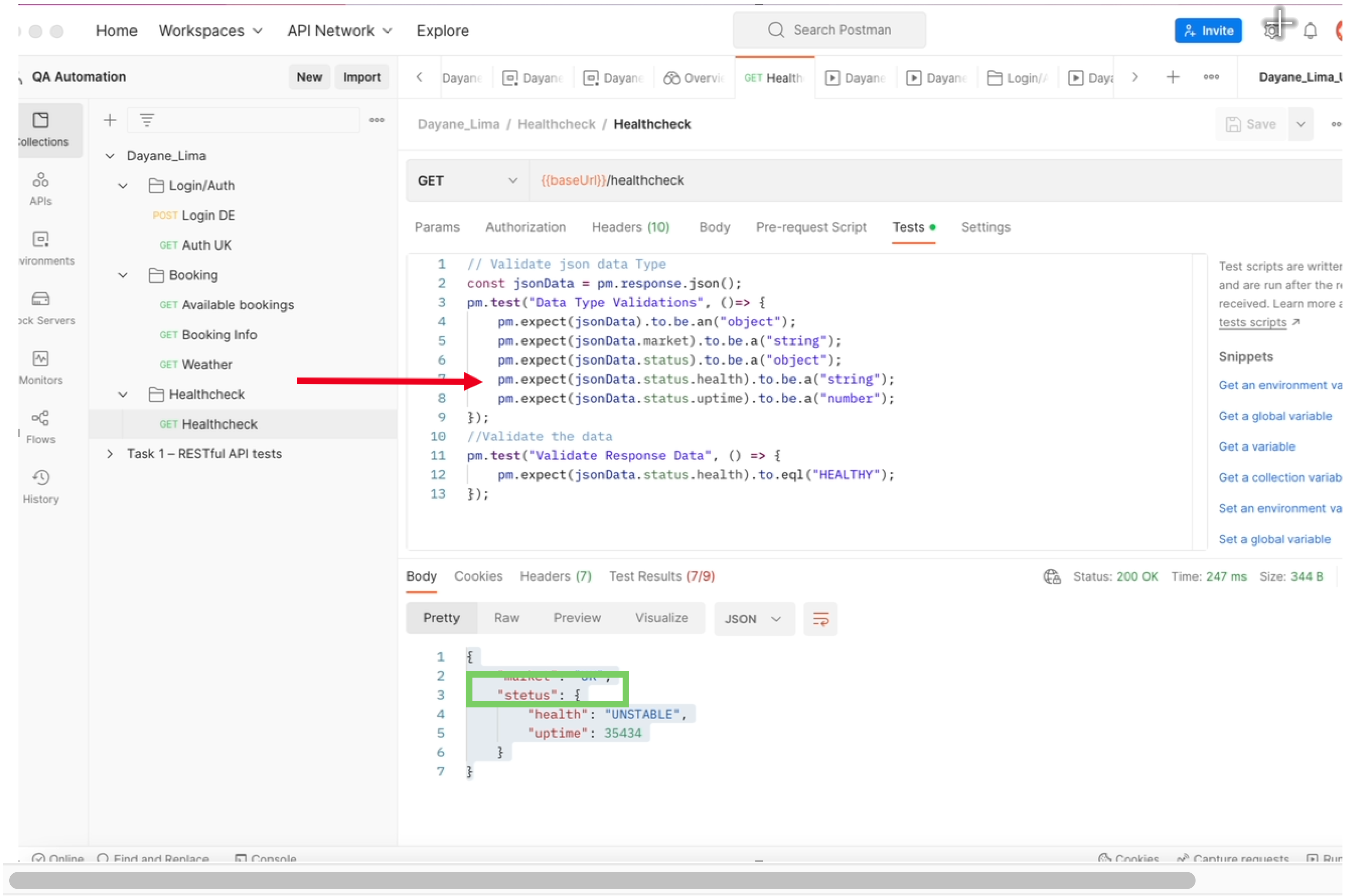
GET {{baseUrl}}/healthcheck

```
// Validate json data Type
const jsonData = pm.response.json();
pm.test("Data Type Validations", ()=> {
    pm.expect(jsonData).to.be.an("object");
    pm.expect(jsonData.market).to.be.a("string");
    pm.expect(jsonData.status).to.be.a("object");
    pm.expect(jsonData.status.health).to.be.a("string");
    pm.expect(jsonData.status.uptime).to.be.a("number");
});
//Validate the data
pm.test("Validate Response Data", () => {
    pm.expect(jsonData.status.health).to.eql("HEALTHY");
});
```

Test scripts are writte and are run after the received. Learn more tests scripts ↗

```
{
    "market": "UK",
    "stetus": {
        "health": "UNSTABLE",
        "uptime": 35434
    }
}
```

Screenshot-8.png 1/1/2023 07:57 PM



GET {{baseUrl}}/healthcheck

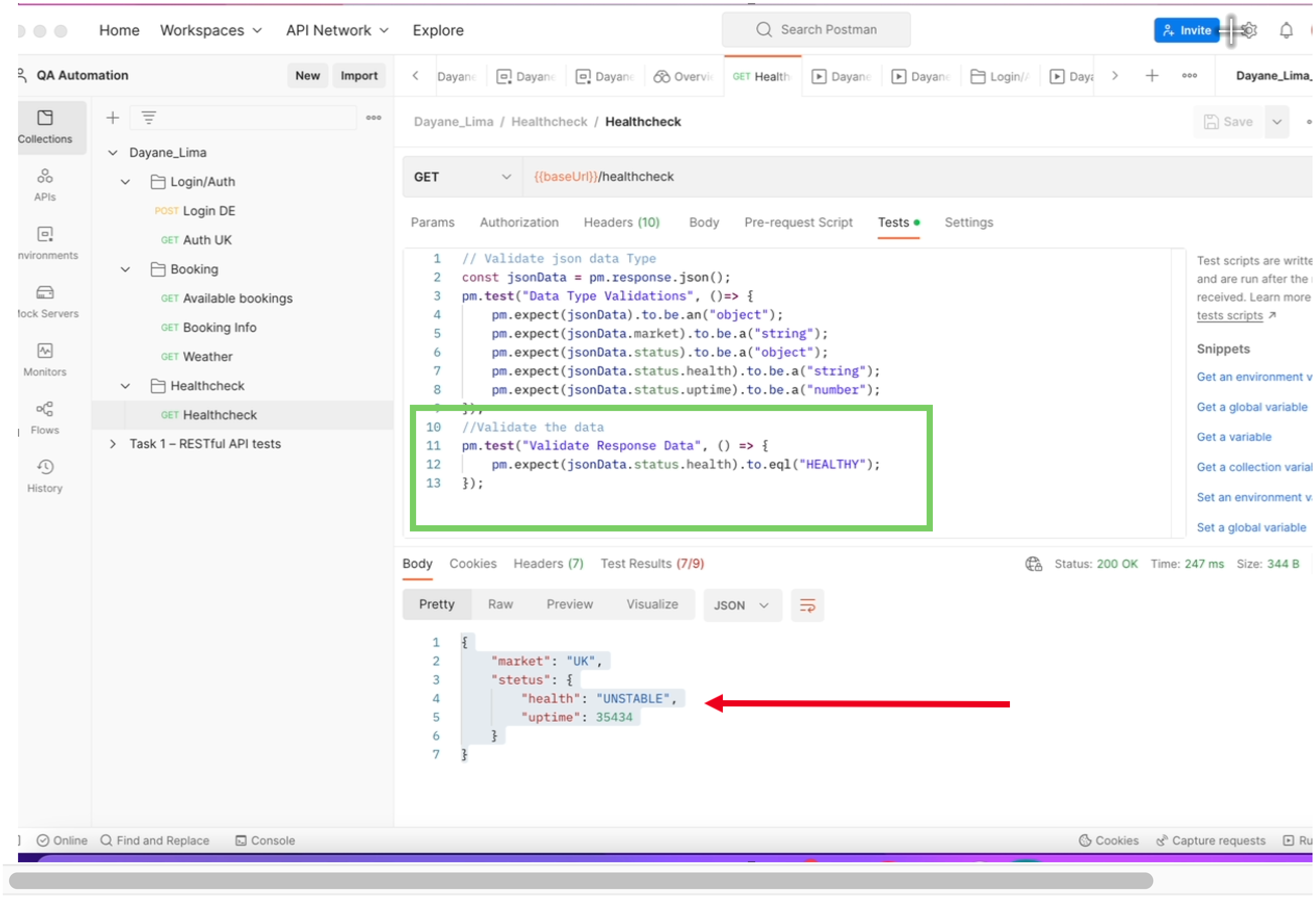
```
// Validate json data Type
const jsonData = pm.response.json();
pm.test("Data Type Validations", ()=> {
    pm.expect(jsonData).to.be.an("object");
    pm.expect(jsonData.market).to.be.a("string");
    pm.expect(jsonData.status).to.be.a("object");
    pm.expect(jsonData.status.health).to.be.a("string");
    pm.expect(jsonData.status.uptime).to.be.a("number");
});
//Validate the data
pm.test("Validate Response Data", () => {
    pm.expect(jsonData.status.health).to.eql("HEALTHY");
});
```

Test scripts are written in JavaScript, and are run after the response is received. Learn more about tests scripts ↗

Status: 200 OK Time: 247 ms Size: 344 B

```
{
    "market": "UK",
    "stetus": {
        "health": "UNSTABLE",
        "uptime": 35434
    }
}
```

Screenshot-9.png 1/1/2023 08:08 PM



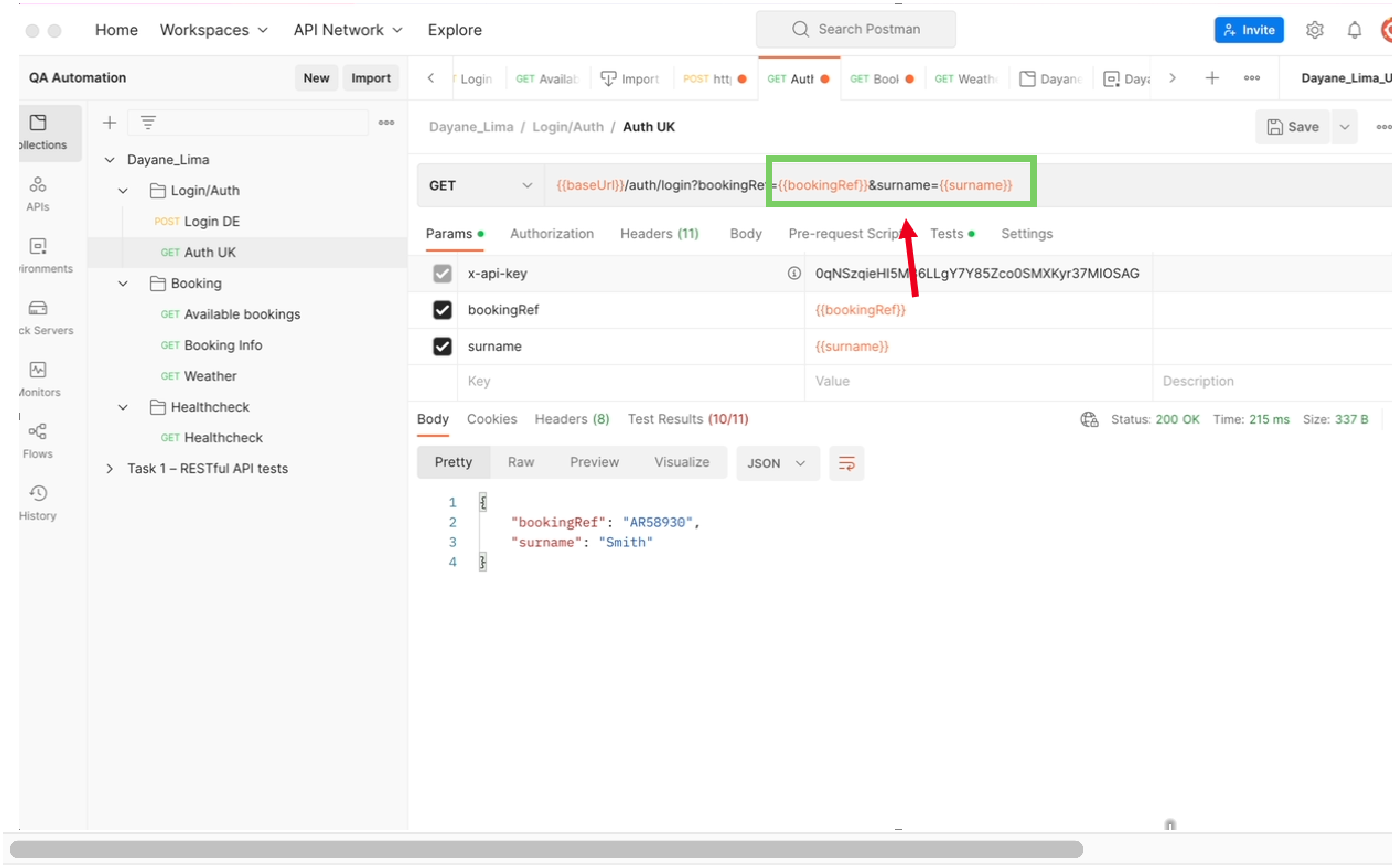
GET {{baseUrl}}/healthcheck

```
// Validate json data Type
const jsonData = pm.response.json();
pm.test("Data Type Validations", ()=> {
    pm.expect(jsonData).to.be.an("object");
    pm.expect(jsonData.market).to.be.a("string");
    pm.expect(jsonData.status).to.be.a("object");
    pm.expect(jsonData.status.health).to.be.a("string");
    pm.expect(jsonData.status.uptime).to.be.a("number");
});
//Validate the data
pm.test("Validate Response Data", () => {
    pm.expect(jsonData.status.health).to.eql("HEALTHY");
});
```

Test scripts are writte and are run after the received. Learn more tests scripts

```
{
    "market": "UK",
    "stetus": {
        "health": "UNSTABLE",
        "uptime": 35434
    }
}
```

Screenshot-10.png 1/1/2023 08:18 PM



Dayane_Lima / Login/Auth / Auth UK

GET {{baseUrl}}/auth/login?bookingRef={{bookingRef}}&surname={{surname}}

	Key	Value	Description
✓	x-api-key	0qNSzqieHI5M?6LLgY7Y85Zco0SMXKyr37MIOSAG	
✓	bookingRef	{{bookingRef}}	
✓	surname	{{surname}}	

Status: 200 OK Time: 215 ms Size: 337 B

```
{
    "bookingRef": "AR58930",
    "surname": "Smith"
}
```

Note-2 1/1/2023 08:22 PM

Environment: Dayane_Lima_UK HTTPS Request: GET https://4s9rh46bpe.execute-api.eu-central-1.amazonaws.com/auth/login?bookingRef=AR58930&surname=Smith The User Authentication information is sent in the URL

End of report.

Want to track bugs in a more seamless manner? Connect to Azure DevOps for free.